iServe: a linked services publishing platform

Conference or Workshop Item

iServe: a Linked Services Publishing Platform

Carlos Pedrinaci, Dong Liu, Maria Maleshkova, David Lambert, Jacek Kopecký, and John Domingue

Knowledge Media Institute, The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
c.pedrinaci@open.ac.uk

Abstract. Despite the potential of service-orientation and the efforts devoted so far, we are still to witness a significant uptake of service technologies outside of enterprise environments. A core reason for this limited uptake is the lack of appropriate publishing platforms able to deal with the existing heterogeneity in the service technologies landscape and able to provide expressive yet simple and efficient discovery mechanisms. In this paper we describe iServe, a novel and open platform for publishing services which aims to better support their discovery and use. It exposes service descriptions as linked data expressed in terms of a simple vocabulary for describing services of different kinds with annotations in diverse formalisms. In addition to describing iServe, this paper also highlights the set of principles behind iServe, which we believe are essential for other generic repositories of semantic information notably ontology repositories.

1 Introduction

Web services are software systems offered over the Internet via platform and programming-language independent interfaces defined on the basis of a set of open standards such as WSDL, SOAP, and further WS-* specifications [1]. The fundamental advantage of Web service technology lies in the support it brings to developing highly complex distributed systems that maximise reuse of loosely coupled components. A key constituent of Service-Oriented Architectures is the service repository, which enables programmatic recording of service descriptions and their subsequent use in the discovery of suitable services. Service publication has therefore been at the core of research and development in this area since the very beginning. However, despite substantial efforts, Web services are not published on the Web in significant numbers, and in practice, lighter and less structured approaches such as Web APIs are currently preferred [2].

One of the main reasons for the paucity of service repositories to date has been the fact that, although they are relatively complex, they do not support expressive queries, limiting their usefulness [3]. Semantic Web Services (SWS) [4] research has devoted considerable efforts to overcoming Web services limitations by enriching them with semantic annotations to better support their discovery, composition and execution. So far, the impact of SWS on the Web has been minimal. In fact, although SWS technologies have already demonstrated benefits, research in the area has glossed over the additional effort demanded of users, and the extra complexity they introduce to the already intricate services technology stack.

Before any significant uptake of services can take place on the Web, better mechanisms for creating, publishing and discovering services must be in place. In particular, service publication must be able to deal with service heterogeneity (e.g., dealing with both WSDL services and Web APIs), it must be based on the use of lightweight semantics able to support relatively advanced yet efficient discovery, and it must be combined with an appropriate set of tools able to support users in the annotation and publication of services.

In this paper we describe iServe, a platform for the seamless publication and discovery of services developed in the context of the EU project SOA4All. iServe addresses the publication of services from a novel perspective based on lessons learned from the evolution of the Web of Data [5]. iServe transforms service annotations expressed in a variety of formats into what we refer to as Linked Services—linked data describing services—that can directly be interpreted by state of the art Semantic Web technologies for their discovery and further processing. The iServe platform is complemented with editors that assist users in creating and publishing service annotations using existing semantic search engines like Watson [6] for searching the Web for reusable ontologies. The decisions adopted for iServe include a number of principles that are of particular relevance for the development of other kinds of repositories for the Web, since they highlight the importance that ontology repositories or systems like Watson may have when it comes to creating Semantic Web applications.

The remainder of this paper is organised as follows. First we cover the state of the art in services description and publication (Section 2). We then introduce iServe and the core principles that underpin the approach (Section 3). Next, we discuss openness as one of the essential characteristics of the approach, and highlight how existing editors and ontology indexing systems can be connected to iServe to better support the creation and publication of service annotations (Section 4). Finally, we present our main conclusions and introduce lines for future research (Section 5).

2 Background and Related Work

The Universal Business Registry part of Universal Description Discovery and Integration (UDDI) [7] is perhaps the best-known effort to support the publication of services on the Web. On the basis of UDDI, large companies like SAP, IBM and Microsoft created a universal registry for enterprise services that could be accessed publicly but it did not gain enough adoption and it was discontinued in 2006 after five years of use. Today, Seekda![1] provides one of the main repositories of publicly available Web services. Seekda's repository currently lists 28,500 Web services with their corresponding documentation, and this number seems to be

[1] See http://webservices.seekda.com/

stagnant. The number of services publicly available contrasts significantly with the billions of Web pages available, and interestingly is not significantly bigger than the 1,500 services estimated to be deployed internally within Verizon [8]. Other academic efforts in crawling and indexing of Web services on the Web have found far lower numbers of services [9].

A major reason for the lack of success of repositories such as UDDI was the fact that, although these registries are relatively complex, they do not support expressive queries, limiting their usefulness [3]. As a consequence, developers did not use these systems since the benefits were often not worth the extra effort. A second reason for the lack of uptake is that Web services have so far essentially targeted enterprises, which as we saw earlier, do not publish services in any significant numbers.

The Web services ecology has recently seen a major evolution with the advent and proliferation of Web APIs and RESTful services [10], and there has not been much progress on, or even concern with, means for describing and discovering these newer kinds of services. Perhaps the most popular directory is ProgrammableWeb.com, which at the time of this writing lists 1,700 APIs and 4,600 mashups. This directory is based on the manual submission of APIs by users and currently provides simple search mechanisms based on keywords, tags, or a simple classification, none of which are particularly expressive. APIHut [11] is a platform that claims to increase the accuracy of keyword-based search of APIs compared to ProgrammableWeb.com or plain Google search, although it does not provide richer discovery mechanisms.

SWS [4] have long tried to overcome the limitations of Web service descriptions by enriching them with semantic annotations. The landscape of SWS is characterised by a number of conceptual models that, despite a few common characteristics, remain essentially incompatible due to the different representation languages and expressivity utilised, as well as because of conceptual differences. Major frameworks include WSMO [12], OWL-S [13], SAWSDL [14], and WSMO-Lite [15]. Regardless of the differences at the semantic level, the vast majority of the SWS initiatives are predicated upon the semantic enrichment of WSDL Web services, and these have turned out not to be prevalent on the Web. Only recently have researchers started focusing on Web APIs and RESTful services, the main examples being SA-REST [16] and MicroWSMO [17].

Enhancing service repositories with semantics and supporting automated discovery has been one of the key topics SWS research has tackled [18–20]. Despite demonstrating advantages of semantic annotations in discovering services, particularly in terms of accuracy and in dealing with heterogeneous data models, SWS work has overlooked the additional complexity involved in creating semantic annotations for services. Consequently, there is no significant body of SWS published in a convenient manner on the Web: the largest public SWS repository today is probably OPOSSum, a test collection with less than 3000 service annotations which provides programmatic access to its content solely through direct access to the database management system [21].

3 Services Publication as Linked Data

The current state of the art evidences limited use of service technologies on a Web scale, and existing technologies for publishing and discovering services remain rather simple, providing limited support and usually based on keyword-based search. This type of mechanism has proven to be insufficient for the needs of software developers, hampering uptake [3]. At the other end of the spectrum, SWS research has aimed for highly advanced discovery techniques but has instead created additional overheads, notably a considerable bottleneck for the creation of rich annotations.

The Web of Data is a relatively recent effort, derived from research on the Semantic Web, whose main objective is to generate a Web exposing and interlinking data previously isolated in silos. The Web of Data is based upon four simple principles, known as the linked data principles, which essentially dictate that every piece of data should be given an HTTP URI which, when looked up, should offer useful information using standards like RDF and SPARQL [5]. Moreover, data should be linked to other relevant resources, thereby allowing humans and computers to discover additional information. Since the linked data principles were outlined in 2006 they have been widely adopted in academic environments, large companies (like the BBC), and national governments (including the United Kingdom's), all of whom are progresively publishing large amounts of data expressed in terms of lightweight ontologies often referred to as vocabularies.

iServe is a novel and open platform for publishing semantic annotations of services based on a direct application of linked data principles to publish service annotations expressed in terms of a simple vocabulary for describing services of different kinds (e.g., WSDL and Web APIs) with annotations in diverse formalisms (e.g., OWL-S, WSMO-Lite). More concretely, iServe is driven by the following conclusions drawn from previous research on service repositories and the progress made by the Web of Data:

- Semantics are essential to reach a minimum level of automation during the life-cycle of services;
- The annotation of services should be simplified in as much as possible;
- On the Web, lightweight ontologies together with the possibility to provide custom extensions prevail against more complex models;
- Any solution to deploying services that aspires to be widely adopted should build upon the various approaches and standards used on the Web, including Web APIs, RDF, and SPARQL;
- Linked data principles are an appropriate means for publishing large amounts of semantic data, both for human and machine consumption;
- Links between publicly available datasets are essential for the scalability and the value of the data exposed.

In the remainder of this section we describe iServe in detail, focusing first on the overall approach and architecture, then describing the ontology it uses, and finally presenting the import mechanisms that make it largely compatible with a wide range of service annotations.

3.1 Overall Approach

iServe supports publishing Linked Services expressed in terms of a simple conceptual model that is suitable for use by both humans and machines, and which abstracts away the existing heterogeneity around service kinds and annotation formalisms. In particular iServe:

- supports importing service annotations in a range of formalisms (e.g., SAWSDL, WSMO-Lite, MicroWSMO) that cover both WSDL services and Web APIs;
- provides means for publishing semantic annotations of services which are automatically assigned a resolvable HTTP URI;
- includes support for content negotiation so that service annotations can be returned in HTML for human users, or in RDF for machine interpretation;
- provides a SPARQL endpoint allowing advanced querying over the services annotations;
- offers a read/write REST API so that services can easily be retrieved and published from remote applications; and
- automatically generates links between the published service annotations and additional documents on the Web such as the original service description or documentation so that users and machines can easily discover more information.

The architecture of iServe, depicted in Figure 1, comprises a crawler, a RESTful API, a set of import mechanisms, and an RDF store. The crawler collects existing annotations from the Web in order to publish them in iServe. Given that not many annotations are published on the Web, the crawler currently deals only with known sets of service annotations such as the SAWSDL retrieval test collection (SAWSDL-TC)[2] and the OWL-S retrieval test collection (OWLS-TC)[3]. The RESTful API, implemented using Restlet[4], provides operations for accessing service annotations and service documentation, allowing remote applications to publish and discover services. The import mechanisms provide support for importing annotations in diverse formalisms by transforming them into the Minimal Service Model, explained in more detail below, which is used by iServe for publishing them as linked data. This service model provides a common vocabulary for service annotations, smoothing away the heterogeneity of different formalisms such as OWL-S, SAWSDL and WSMO, allowing humans and machines to discover service annotations originally described using heterogeneous conceptual models through a single vocabulary. Finally, iServe captures the service annotations together with some provenance information including the annotation author and the creation or modification date in an RDF store. RDF storage and querying support is provided by Swift OWLIM,[5] although it is accessed via RDF2Go[6] to maintain independence with respect to the concrete store used. The RDF

[2] See http://www.semwebcentral.org/projects/sawsdl-tc/
[3] See http://www.semwebcentral.org/projects/owls-tc/
[4] See http://www.restlet.org/
[5] See http://www.ontotext.com/owlim/
[6] See http://semanticweb.org/wiki/RDF2Go

store provides us with a SPARQL endpoint that is made available to external applications in order to interact with iServe for retrieving services.

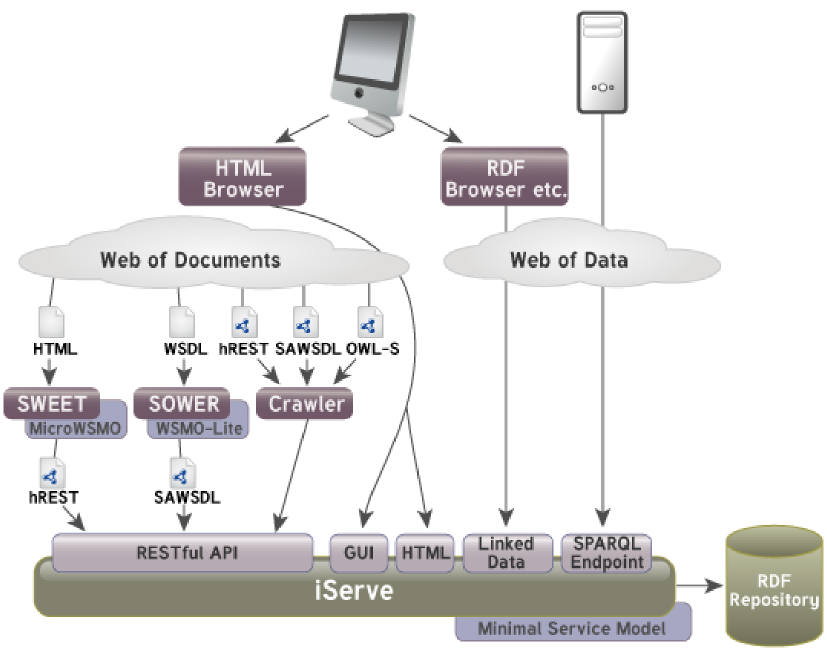

Fig. 1. The overall architecture of iServe.

3.2 Conceptual Models

Section 2 covered the state of the art around the publication of services and semantic annotations, highlighting the existing heterogeneity in terms of languages and formalisms used, the main approaches adopted thus far for publishing and discovering services, and the need of finding a trade-off between expressivity of service descriptions and their complexity. In building a system like iServe it is necessary to provide a common vocabulary, able to describe services in a way that allows machines to automatically locate and filter services according to their functionality or the data they handle, independent of the formalism originally used to describing them.

The best-known approaches to annotating services semantically are OWL-S, WSMO, SAWSDL, and WSMO-Lite for WSDL services, and MicroWSMO, and SA-REST for Web APIs. To cater for interoperability, iServe uses what can essentially be considered the largest common denominator between these formalisms which we refer to as the Minimal Service Model (MSM). The MSM, first introduced together with WSMO-Lite [15], is a simple RDF(S) ontology able to capture the core semantics of both Web services and Web APIs in a common model supporting the common publishing and search of services, yet still permitting framework-specific extensions to remain attached and thereby benefitting those clients able to comprehend those formalisms.

The MSM, denoted by the `msm` namespace in Figure 2, defines `Service`s which have a number of `Operation`s. `Operation`s in turn have input and output `MessageContent` descriptions, and `Fault`s. A `MessageContent` may be composed of `MessagePart`s which can be mandatory or optional. The addition of message parts extends the earlier definition of the MSM as described in [15]. The intent of the message part mechanism is to support finer-grain discovery based on message parts, mirroring the granularity of SAWSDL and allowing to distinguish between mandatory and optional parts.

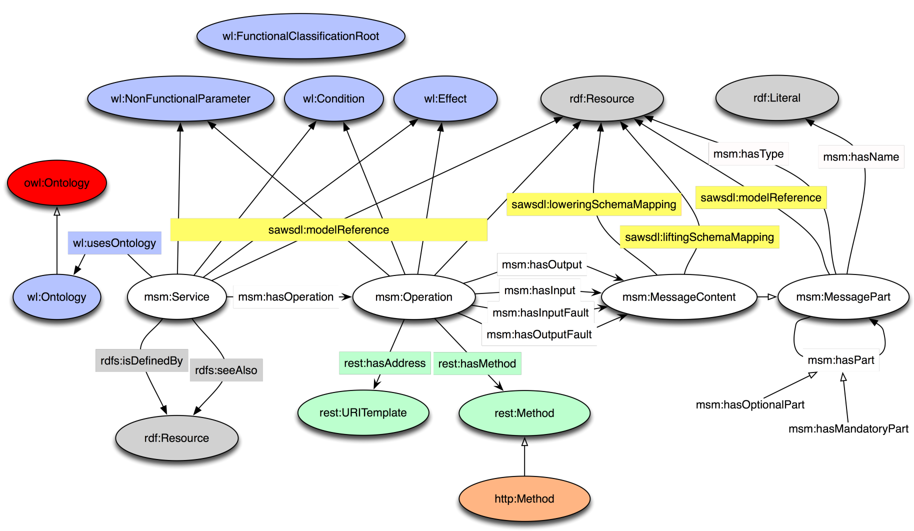

Fig. 2. Conceptual model for services used by iServe.

iServe also uses the SAWSDL, WSMO-Lite and hRESTS vocabularies, depicted in the figure with the `sawsdl`, `wl`, and `rest` namespaces respectively. The SAWSDL vocabulary captures in RDF the three kinds of annotations over WSDL and XML Schema, namely `modelReference`, `liftingSchemaMapping` and `loweringSchemaMapping` that SAWSDL supports. SAWSDL supports the annotation of WSDL and XML Schema syntactic service descriptions with semantic concepts, but does not specify a particular representation language nor does it provide any specific vocabulary that users should adopt. The `modelReference` property links syntactic service elements to semantic models via URIs, while the schema mapping properties indicate data transformations from a syntactic representation to its semantic counterpart and vice versa.

WSMO-Lite builds upon SAWSDL by extending it with a model specifying the semantics of the particular service annotations. It provides a simple RDFS ontology together with a methodology for expressing functional and non-functional semantics, and an information model for WSDL services based on SAWSDLs modelReference hooks. In particular, WSMO-Lite makes explicit the intended meaning for `modelReference` annotations without modifying SAWSDL. Instead, it provides a vocabulary to annotate the URIs pointed to by SAWSDL annotations. The WSMO-Lite vocabulary includes the classes `NonFunctionalParameter`, `FunctionalClassificationRoot`, `Condition`, `Effect` and `Ontology`. With these, an annotator can type SAWSDL annotation references without adding information directly to the WSDL description.

The hRESTS vocabulary [22] extends the MSM with specific attributes for operations to model information particular to Web APIs, such as a `URITemplate` to describe the URI for invocation, and `method` to indicate the HTTP method used for the invocation. For methods, iServe uses the draft W3C HTTP in RDF vocabulary [23].

4 An Open Publishing Platform

The fundamental objective pursued by iServe is to provide a platform able to publish service annotations in a way that would allow people to achieve a certain level of expressivity and refinement in discovering services, while remaining simple and convenient both for human and machine use. The simple conceptual model explained earlier is a principal building block to support this as a general model able to abstract away the existing conceptual heterogeneity among service annotation approaches without introducing considerable complexity from a knowledge acquisition and computation perspectives. Thanks to its simplicity, the MSM captures the essence of services in a way that can support service matchmaking and invocation and still remains largely compatible with the RDF mapping of WSDL [24], with WSMO-based descriptions of Web services, with OWL-S services, and with services annotated according to WSMO-Lite and MicroWSMO. Although providing a formal mapping for each of these languages is out of the scope of this paper, we note that the elements captured in the MSM are largely common to existing models. The mapping is not lossless, but appropriate use of `rdfs:isDefinedBy`, covered next, can help circumvent this limitation and still provide a common ground for publishing Linked Services in the Web of Data in a way that is amenable to automated processing and where more expressive definitions can be linked if needed.

iServe provides a set of import mechanisms that can take service annotations in several formats, and generate the appropriate RDF in terms of iServe's conceptual model. In particular, the current version can import SAWSDL, WSMO-Lite, MicroWSMO, and part of OWL-S descriptions. The import process generates `rdf:isDefinedBy` links from the service annotation to the original description file (e.g., the WSDL including the annotations), `rdf:seeAlso` to any documentation about the service (e.g., the Web API description page itself), and `owl:sameAs` relations to the published version of the same service annotation in other systems. Currently, the only `owl:sameAs` links that are generated link to the RDF mapping of WSDL [24], ensuring compatibility with tools using it, but in the future, other links could be generated to connect services stored in other repositories adopting principles similar to iServe's.

In addition to the import mechanisms that provide compatibility and abstract away the heterogeneity in service description formalisms, iServe contributes to publishing services on the Web by automatically hosting service descriptions at a publicly accessible URI, and by offering means for users or machines to upload service annotations through its RESTful API[7].

iServe is integrated with two service annotation editors being developed within the SOA4All project: SWEET [25] (SemanticWeb sErvices Editing Tool) and SOWER (SWEET is nOt a Wsdl EditoR), which support users in annotating Web APIs and WSDL services respectively.[8] Both editors are Web applications that can directly be used with a Web browser and provide support for browsing service descriptions and annotating them through a simple interface. Behind the scenes, both editors take care of serialising the descriptions according to MicroWSMO (SWEET) and WSMO-Lite (SOWER) specifications respectively. Both editors strive to assist users in creating service annotations through two main means: the use of the Web as background knowledge and a direct connection with iServe for the seamless persistence and publication of annotations.

A fundamental part of service annotation concerns the linking of (parts of) service descriptions to ontologies capturing semantically the data model they handle, certain non-functional properties such as the quality of service and price, and their categorisation with respect to reference taxonomies and classifications of services. Thus far, the authoring of service annotations has essentially been based on the manual creation of both the services' structure and the actual ontologies used in their annotation. Consequently, creating service annotations has been a particularly tedious task which rather than better supporting the integration of services has led instead to additional heterogeneity at the semantic level. Further limitations have also traditionally been brought by the fact that the service annotations and the ontologies used were most often not published publicly which impedes the interpretation and use of services by third-parties.

SWEET and SOWER support the integrated search of suitable domain ontologies by means of Watson [6]. Watson serves as a gateway for the Semantic Web by collecting the available semantic content on the Web, analyzing it to extract useful metadata and generate indexes, and implementing efficient query facilities to access the semantic data. During the annotation of services, Watson supports retrieving ontologies and concepts matching particular keywords. In this way, a user can select a service property and receive a list of semantic entities

[7] More details about the API can be found at http://iserve.kmi.open.ac.uk
[8] A publically accessible SWEET instance is available at http://sweet.kmi.open.ac.uk.

suitable for the annotation thus potentially reducing the manual labour involved through reuse (Figure 3). Integrating the editors with systems like Watson gives annotators better access to semantic vocabulary, hopefully leading to descriptions that are both more precise, and more widely understood by dint of being found on the Web and reused rather than invented on an ad-hoc basis, thereby embedding service descriptions in the existing Web of linked data.

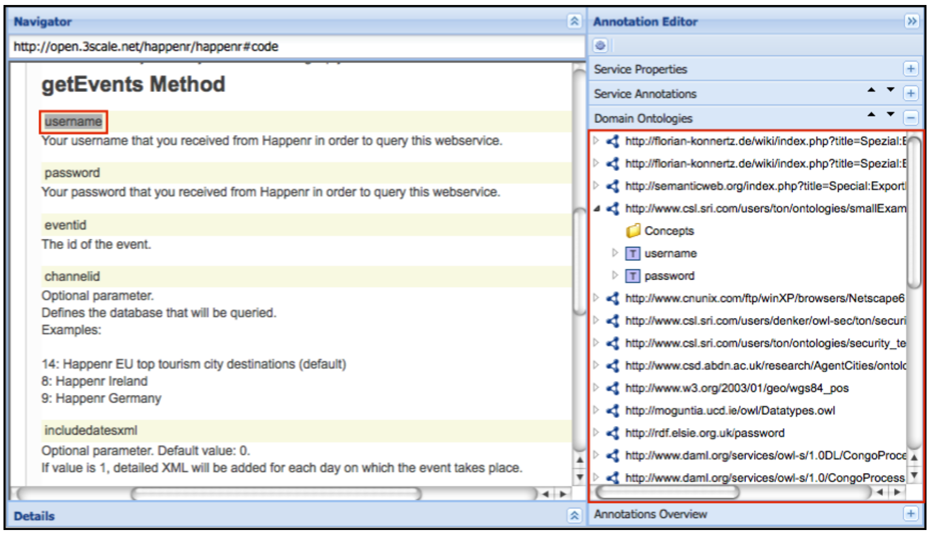

Fig. 3. SWEET showing possible annotations based on Watson results.

5 Conclusions and Future Work

Despite the potential of service technologies and the efforts devoted so far, we have yet to witness a significant uptake of service technologies outside enterprise environments. There is a tension between the rich descriptions necessary to automate much of the use of services, and the reticence on the part of developers to invest that effort. A similar situation exists in Semantic Web Services, where the expressive models able to capture the semantics of services stand in contrast to the lightweight tagging approaches that appear more acceptable to users.

In this paper we described iServe, a novel and open platform for publishing services that aims to better support their discovery and use by exposing them as linked data expressed in terms of a simple vocabulary for services. This vocabulary provides a common ground for descriptions of different flavours of Web services (SOAP services described in WSDL, and RESTful services described in hRESTS), and several kinds of semantic formalisms (including OWL-S, WSMO, and WSMO-Lite). iServe is underpinned by principles we believe are applicable to generic repositories of semantic information. These principles include the application of linked data principles to publish metadata about the content, the automated publication of the actual content at a public URI if necessary, the adoption of a simple vocabulary able to abstract away the heterogeneous vocabularies or formalisms used by other repositories within the domain, and the provisioning of an open API allowing applications to automatically retrieve and publish information.

Additionally in this paper we have shown how the advent of engines like Watson that support searching for ontological entities over the Web has enabled a new way for authoring semantic models which as we have seen can also contribute to authoring service annotations by reducing the amount of effort required and potentially contributing to reducing integration issues through ontology reuse.

Future work on iServe will focus on the development and refinement of import mechanisms for WSMO and OWL-S, the creation of advanced indexing mechanisms, and capturing non-functional information about services gleaned at runtime from the monitoring infrastructure. Future work on the editor side will focus on better assisting users in the creation of annotations.

Acknowledgements This work was partly funded by the EU project SOA4All (FP7-215219). We thank all the members of the SOA4All project and the Conceptual Models for Services Working Group of STI International. We also thank Pierre Grenon for his insightful comments and Alex Simov for the development of SOWER.

References

1. Erl, T.: SOA Principles of Service Design. The Prentice Hall Service-Oriented Computing Series. Prentice Hall (July 2007)
2. Davies, J., Domingue, J., Pedrinaci, C., Fensel, D., Gonzalez-Cabero, R., Potter, M., Richardson, M., Stincic, S.: Towards the open service web. BT Technology Journal **26**(2) (2009)
3. Pilioura, T., Tsalgatidou, A.: Unified publication and discovery of semantic web services. ACM Trans. Web **3**(3) (2009) 1–44
4. McIlraith, S., Son, T., Zeng, H.: Semantic web services. IEEE Intelligent Systems **16**(2) (March 2001) 46–53
5. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) (2009)
6. d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Toward a new generation of semantic web applications. IEEE Intelligent Systems **23**(3) (2008) 20–28
7. Clement, L., Hately, A., von Riegen T. Rogers, C.: UDDI Specification Version 3.0.2. Technical report, OASIS (2004)
8. Stollberg, M.: Scalable Semantic Web Service Discovery for Goal-driven Service-Oriented Architectures. PhD thesis, Faculty of Mathematics, Computer Science and Physics Leopold-Franzens University Innsbruck, Austria (March 2008)
9. Al-Masri, E., Mahmoud, Q.H.: Investigating web services on the world wide web. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 795–804

10. Richardson, L., Ruby, S.: RESTful Web Services. O'Reilly Media, Inc. (May 2007)
11. Gomadam, K., Ranabahu, A., Nagarajan, M., Sheth, A.P., Verma, K.: A faceted classification based approach to search and rank Web APIs. In: ICWS '08: Proceedings of the 2008 IEEE International Conference on Web Services, Washington, DC, USA, IEEE Computer Society (2008) 177–184
12. Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J.: Enabling Semantic Web Services: The Web Service Modeling Ontology. Springer (2007)
13. Martin, D., Burstein, M., J., H., Lassila, O., McDermott, D., McIlraith, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services. http://www.daml.org/services/owl-s/1.0/owl-s.pdf (2004)
14. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema. http://www.w3.org/TR/sawsdl/ (January 2007) W3C Candidate Recommendation 26 January 2007.
15. Vitvar, T., Kopecky, J., Viskova, J., Fensel, D.: WSMO-Lite annotations for web services. In Hauswirth, M., Koubarakis, M., Bechhofer, S., eds.: Proceedings of the 5th European Semantic Web Conference. LNCS, Berlin, Heidelberg, Springer Verlag (June 2008)
16. Sheth, A.P., Gomadam, K., Lathem, J.: SA-REST: Semantically interoperable and easier-to-use services and mashups. IEEE Internet Computing **11**(6) (2007) 91–94
17. Maleshkova, M., Kopecký, J., Pedrinaci, C.: Adapting SAWSDL for semantic annotations of restful services. In: Workshop: Beyond SAWSDL at OnTheMove Federated Conferences & Workshops. (2009)
18. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. Web Semantics: Science, Services and Agents on the World Wide Web **1**(1) (2003) 27 – 46
19. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with OWLS-MX. In: AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM (2006) 915–922
20. Stollberg, M., Hepp, M., Hoffmann, J.: A caching mechanism for semantic web service discovery. In: 6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007). (November 2007) 477–490
21. Küster, U., König-Ries, B.: Towards standard test collections for the empirical evaluation of semantic web service approaches. Int. J. Semantic Computing **2**(3) (2008) 381–402
22. Kopecky, J., Gomadam, K., Vitvar, T.: hRESTS: an HTML Microformat for Describing RESTful Web Services. In: The 2008 IEEE/WIC/ACM International Conference on Web Intelligence (WI2008), Sydney, Australia, IEEE CS Press (November 2008)
23. Koch, J., Velasco, C.A.: HTTP vocabulary in RDF 1.0. Working draft, W3C (October 2009)
24. Kopecký, J.: Web services description language (WSDL) version 2.0: RDF mapping. Working group note, W3C (June 2007)
25. Maleshkova, M., Pedrinaci, C., Domingue, J.: Supporting the creation of semantic RESTful service descriptions. In: Workshop: Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2) at 8th International Semantic Web Conference. (2009)